STM JOURNALS

Journal of
Computer Technology & Applications

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 15, Issue 3, 2024
September–December
DOI (Journal): 10.37591/JoCTA

https://journals.stmjournals.com/jocta

Review JoCTA

Data Recovery Using Brute Force Algorithm: A Review

Shifa Thamkin S.[1,*], Raksha H.M.[2], Mohammed Rafi[3]

Abstract

This article delves into the intriguing field of data recovery, with a particular emphasis on the brute force approach, a powerful yet frequently underappreciated method. We begin by exploring the evolution of data recovery techniques since the 1970s and 1980s, examining traditional methods used to retrieve lost or corrupted data. Following this historical perspective, the article provides a brief overview of various methodologies employed in data recovery, including techniques that leverage brute force. Brute force approaches, while often time-consuming, can be highly effective in recovering lost data, particularly when other methods fall short. The article further compares brute force techniques with more modern data recovery methods, such as those utilizing machine learning algorithms, to highlight their relative strengths and weaknesses. In addition, we discuss potential future advancements in the field, focusing on improving computational efficiency and integrating brute force methods with emerging technologies like quantum computing and parallel processing. These advancements hold the promise of making data recovery faster and more accurate, thus broadening the application of brute force in critical situations. Ultimately, this article aims to provide a comprehensive overview of data recovery techniques, with a forward-looking perspective on how brute force may evolve in the context of technological progress.

Keywords: Data recovery, brute force, time complexity, space complexity, corrupted data, machine learning, quantum computing

INTRODUCTION

Data recovery is the process of restoring data that has been lost, accidentally deleted, corrupted, or rendered inaccessible. This process involves accessing and retrieving data that have become unreachable owing to reasons such as accidental deletion, formatting, hardware failures, software issues, virus attacks, or physical damage to the storage device [1]. The term "data recovery" can be categorized into two main contexts: personal data recovery and forensic data recovery. Personal data recovery typically refers to retrieving data that has been unintentionally lost or made inaccessible, such as from damaged storage media. In contrast, forensic data recovery focuses on recovering data that has been deliberately encrypted or concealed to prevent access by others, including forensic investigators [2].

***Author for Correspondence**
Shifa Thamkin S.
E-mail: utstshifa@gmail.com

[1,2]Student, Department of Computer Science and Engineering, University BDT College of Engineering, Davangere, Karnataka, India
[3]Professor, Department of Computer Science and Engineering, University BDT College of Engineering, Davangere, Karnataka, India

Received Date: August 14, 2024
Accepted Date: September 23, 2024
Published Date: October 07, 2024

Citation: Shifa Thamkin S., Raksha H.M., Mohammed Rafi. Data Recovery Using Brute Force Algorithm: A Review: Review. Journal of Computer Technology & Applications. 2024; 15(3): 10–16p.

Brute-force algorithms are straightforward and systematic methods for data recovery. They operate by methodically testing each possible combination within a defined search space until they identify the correct data. Imagine searching for a lost key by trying every key on your keychain one by one [3]. The core principle behind brute-force algorithms is an exhaustive search. These involve generating and testing all possible solutions, even if some solutions are highly improbable. While brute force may sound

simplistic, it can be remarkably effective, particularly in scenarios where the search space is relatively small, or the data being sought is highly unique. Brute-force algorithms, characterized by their exhaustive search approach, are often employed when other methods fail or are impractical [4].

This article aims to elucidate the principles behind brute-force data recovery, evaluate its efficiency, and place it within the broader context of data recovery solutions.

Some of the essential elements of the structured data recovery process are as follows:

1. *Identify the target data:* Determine the type and structure of the lost data.
2. *Generate candidates:* Create all possible combinations of data blocks or keys.
3. *Validation:* Check each candidate against predefined criteria to determine whether it reconstructs the original data [5, 6].

LITERATURE SURVEY

In the realm of data recovery, brute-force methods have been utilized since the 1970s and the 1980s, particularly in password recovery and file reconstruction. Early works, such as those by Stallings (1982) [7], highlighted the viability of brute-force approaches despite their computational intensity [7]. Password recovery is one of the most well-documented applications of brute force algorithms.

The seminal work by Schneier (2007) in "Applied Cryptography" provided an in-depth analysis of brute force attacks on cryptographic systems [8]. This book provides a comprehensive overview of cryptographic techniques, including brute-force methods for password recovery. Schneier's work demonstrated how brute force can systematically crack passwords by iterating through all possible combinations. The time complexity for brute-force password recovery is $O(k^n)$ [8].

File carving, which involves recovering files without relying on file system metadata, has seen significant contributions from researchers, such as Garfinkel (2007) and Richard et al. (2005). Garfinkel's "Carving Contiguous and Fragmented Files with Fast Object Validation" introduced techniques to improve the efficiency of brute force methods in file carving [9], while Richard's et al. research on "Scalpel: A Frugal, High Performance File Carver" showcased a practical tool that leveraged brute force techniques for efficient data recovery [10]. The time complexity for file carving using brute force can be approximated as $O(n^k)$, where n is the number of data blocks, and k is the average size of the files. The space complexity is $O(n)$ because it involves storing the data blocks being processed [11].

Handling Corrupted Data

The recovery of corrupted data using brute-force methods has been explored in works such as those by Datar and Muthukrishnan (2002) [12]. These studies focused on the algorithmic foundations required to piece data fragments together, employing brute force in conjunction with other techniques, such as redundancy checks and error correction codes. A hybrid approach typically reduces the search space, resulting in improved performance. The time complexity for recovering corrupted data is generally $O(kn)$, where k is the number of possible values for each corrupted piece and n is the number of corrupted pieces. Space complexity can differ but is commonly O(n) because it requires storing intermediate recovery attempts.

Brute-force techniques for data recovery have become increasingly sophisticated by leveraging advances in hardware, software, and cloud computing. Recent literature highlights that although brute-force attacks remain a straightforward approach, they are becoming more feasible and dangerous owing to advancements in technology [13].

Hardware and Cloud Utilization

Modern brute-force attacks often employ high-performance GPUs, such as Nvidia RTX 4090, which can process billions of password guesses per second. In addition, attackers are now using cloud services

to access multiple virtual GPUs simultaneously, making the brute-force method faster and more accessible than ever before [14].

Automation and AI

AI further accelerates brute-force attacks by optimizing guessing strategies, significantly reducing the time needed to crack even complex passwords. Research indicates that AI-driven tools can crack a significant portion of passwords within a few seconds or minutes, depending on the strength and complexity of the password [14].

Hashing Algorithms and Password Security

Despite the implementation of secure hashing algorithms, older methods, such as MD5, remain vulnerable. Research shows that brute-force attacks can still crack passwords protected by outdated hashing methods, particularly when those passwords are reused across multiple platforms [15].

METHODOLOGY

This section explores different methods of data recovery in different contexts, such as password recovery, file carving, and corrupted data recovery.

Password Recovery

In password recovery, the brute-force method entails creating and testing every conceivable password combination until the correct one is identified.

1. Identify the Character Set
   - Determine the possible characters in the password (e.g., lowercase letters, uppercase letters, digits, and special characters).
   - Example: In a simple case, if the password includes only lowercase letters, the character set is {a, b, c, ..., z}.
2. Set Password Length Range
   - Specify the shortest and longest lengths allowed for the password.
   - Example: If the password length is between four and six characters, we set this as the range.
3. Generate Combinations
   - Systematically generate every possible combination of characters within the defined length range.
   - Use nested loops or recursion to create all possible strings.
   - Example: For length 4 and character set {a, b}, the combinations are *aaaa, aaab, aaba, aabb, abaa, abab, abba, abbb, baaa, baab, baba, babb, bbaa, bbab, bbba, bbbb.*
4. Test Combinations
   - Attempt to access the account or decrypt the data with each generated password.
   - Validate if the combination successfully recovers the data or provides access.
   - Example: For each generated password, log in or decrypt the file until the correct password is found.

Time Complexity

The time complexity is O(kn), where k represents the size of the character set and n denotes the length of the password.

Space Complexity

The space complexity is *O(1)* because only one password must be stored simultaneously during the testing process. Password recovery can be achieved by applying brute force using "*ProcessPoolExecutor*" in Python [16].

File Carving

File carving is a technique used in computer forensics to retrieve data from a disk drive or other storage device without relying on the original file system that created the file. This method recovers

files from an unallocated space without file metadata and is employed for data recovery and digital forensic investigations. Also known as "carving," this term broadly refers to extracting structured data from raw data based on format-specific characteristics found in the structured data [17].

The difference between file recovery and file carving is that file recovery techniques make use of file system information, whereas file carving works only on raw data on the media and is not connected with the file system structure [17].

Tools for File Carving

- *Foremost:* Command-line tool that is widely used for file carving. It searches for the file headers and footers defined in the configuration files.
- *Scalpel:* A more efficient and faster version of foremost is used to carve out specific file types [18].

Steps to Perform Brute Force File Carving

1. *Obtain a disk image:* Create an image of the data source (e.g., dd command in Linux).
2. *Select the carving tool:* Choose a tool like Foremost or Scalpel.
3. *Configure the tool:* Define the file types to search (based on signatures).
4. *Run the tool:* Execute the tool to scan the image and recover files.
5. *Analyze results:* Review the recovered files and validate their integrity.

Time complexity: The time complexity is O(nk), where n represents the number of data blocks and k denotes the average size of the files.

Space complexity: The space complexity is O(n) because it involves storing the data blocks that are being processed.

Corrupted Data Recovery

Data corruption involves errors in computer data that arise during writing, reading, storage, transmission, and processing, leading to unintended alterations in the original data. Unlike straightforward data loss, corrupted data remains in the system but is altered, incorrect, or unusable.

Corrupted data can be recovered using brute force by the below-mentioned steps:

1. Prepare the Environment
   - *Backup:* Create a backup of the corrupted disk or data source before attempting recovery. This is essential for preventing additional data loss. Disk Image: Create a disk image of a corrupted storage device using tools such as dd in Linux. Using an image file prevents additional damage to the original data.
2. Select a File Carving Tool
   - Choose a file carving tool suitable for brute force recovery. Popular tools include:
     - *Foremost:* A widely used open-source tool for carving out files based on header and footer signatures.
     - *Scalpel:* A high-speed version of Foremost, known for its efficiency.
     - *Photorec:* Another powerful tool for recovering files by scanning disk images.
3. Configure the Tool
   - *File-type signatures:* Configure the tool to search for specific file types by defining their signatures. Most tools include predefined signatures for common file types but allow customization.
   - *Output Directory:* Specify the directory where the recovered files will be saved.
4. Run the Carving Process
   - The file carving tool is executed on the disk image. The tool scans the image byte-by-byte, searches for file signatures, and reconstructs files based on the identified signatures.

5. Analyze and Validate Recovered Files
   - After the carving process, review the recovered files in the output directory.
   - *Check File Integrity:* Validate the integrity of recovered files using file viewers or specialized tools. This step is crucial because some files might be partially recovered or corrupted.
   - *File Fragmentation:* If files are fragmented, they may need to be reassembled manually or with tools designed for fragment recovery.
6. Document of the Process
   - Detailed records of the steps taken during recovery, particularly in forensic or legal contexts. Appropriate documentation ensures that the recovery process is repeatable and verifiable.
7. Restore or Rebuild the Data
   - Once the files are recovered and validated, they are restored to a new storage device. Avoid using the original corrupted storage to prevent further issues.
8. Brute Force vs. Machine Learning
   - Machine learning techniques can predict and recover data based on patterns and prior knowledge, offering faster recovery times, but requiring extensive training and data preprocessing. Brute force does not rely on prior knowledge and can be used universally, albeit at a higher computational cost.

COMPARATIVE STUDIES

Numerous comparative studies have evaluated brute-force algorithms against other data-recovery methods. For instance, Altheide and Carvey's "Digital Forensics with Open-Source Tools" (2011) [15] contrasted brute force approaches with heuristic and signature-based methods, highlighting the trade-offs in terms of speed, accuracy, and applicability. Similarly, Böhme and Freiling (2008) [19] in "On metrics and measurements" provided a thorough analysis of the computational costs associated with brute force, emphasizing the need for more efficient algorithms in large-scale data recovery.

The rise of high-performance computing and parallel processing has reinvigorated the interest in brute-force algorithms. Research by Goodrich and Tamassia (2011) [20] in "Algorithm Design: Foundations, Analysis, and Internet Examples" discussed how modern computing infrastructure could be leveraged to enhance brute force techniques. The advent of quantum computing promises to revolutionize brute-force data recovery by significantly reducing the time complexity of these algorithms.

For instance, Grover's algorithm can reduce the time complexity of the brute force search to $O(\sqrt{N})$, where N is the number of possible solutions, significantly speeding up the process compared with classical brute force methods.

Several AI-driven methods contribute to data recovery, such as machine learning, data processing, and predictive analysis. AI-driven methods have proven to be more efficient and adaptable and have improved accuracy.

AI-driven recovery uses machine-learning algorithms to identify patterns and predict the location of files. This allows for more targeted scanning, which can significantly reduce the time required for recovery, whereas brute force methods do not prioritize or optimize the scanning process, leading to potential inefficiencies when dealing with large or complex datasets.

AI-driven data recovery represents a significant advancement over traditional methods, as it offers enhanced accuracy, adaptability, and efficiency. By leveraging machine learning and deep learning techniques, these systems can handle complex recovery scenarios, making them particularly valuable in environments where data integrity is critical.

HYBRID APPROACHES AND FUTURE DIRECTIONS

Recent studies have focused on integrating brute force with other advanced techniques. For example, Ramkumar et al. (2022) [21] in "Codes for Distributed Storage" explored how combining brute force

with machine learning and heuristic methods can improve the efficiency and effectiveness of data recovery processes.

The future of brute-force data recovery lies in enhancing computational efficiency and integrating it with other advanced techniques. Potential advancements include the following.

- *Quantum computing:* Leveraging quantum algorithms to significantly speed up brute force searches.
- *Hybrid approaches:* Combining brute force with heuristic or machine learning methods to balance speed and accuracy.
- *Parallel processing:* Distributed computing resources are utilized to parallelize brute force operations and reduce the recovery time.

AI-driven data recovery methods are generally superior to brute-force methods in most scenarios because of their efficiency, accuracy, adaptability, and ability to handle complex recovery tasks. While brute-force methods are still useful, particularly in straightforward recovery scenarios where the data are not heavily corrupted or fragmented, AI-driven approaches offer significant advantages in more challenging environments. The trade-off is that AI-driven methods require more computational resources and an initial setup (training data), but their ability to adapt and improve over time makes them a powerful tool for modern data recovery.

CONCLUSION

Brute-force algorithms remain a fundamental tool in data recovery. Although they have inherent limitations in terms of speed and resource usage, advancements in computing technology and new revolutionary methodologies provide hope for enhancing their efficiency. As data continues to grow in volume and complexity, the role of brute force in data recovery will undoubtedly evolve, maintaining its relevance in the ever-changing landscape of data management. Though there are several AI-driven data recovery methods, as the world witnesses the revolution of AI there may be more advanced data recovery methods in the future which can be associated with brute force or other methodologies to provide more efficient and accurate methods for data recovery.

REFERENCES

1. GeeksforGeeks. (2024). What is data recovery? [online] Available from: https://www.geeksforgeeks.org/what-is-data-recovery/.
2. Boulos D, Zemljic N. (2024). Data recovery. [online] Available from: www.cs.toronto.edu/~arnold/427/15s/csc427/indepth/dataRecovery/DataRecovery_writeup.pdf.
3. File carving. Infosec. Infosecinstitute.com. [online] Available from: https://www.infosecinstitute.com/resources/digital-forensics/file-carving/. 2018.
4. Alkhwaja I, Albugami M, Alkhwaja A, Alghamdi M, Abahussain H, Alfawaz F, Almurayh A, Min-Allah N. Password cracking with brute force algorithm and dictionary attack using parallel programming. Appl Sci. 2023;13:5979. DOI: 10.3390/app13105979.
5. Fields-data-recovery. (2021). How to recover lost data: A step-by-step guide. Fields-data-recovery.com. [online] Available from: https://www.fields-data-recovery.com/blog/how-to-recover-lost-data-a-step-by-step-guide.
6. Garry. (2023). How do you create a data recovery plan? – Darwin's Data. [online] Available from: https://darwinsdata.com/how-do-you-create-a-data-recovery-plan/.
7. Stallings W. Cryptography and Network Security Principles and Practice, 7th edition. Global edition. SPi Global. Pearson. Malaysia. [online] Available from: https://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network-security_-principles-and-practice-7th-global-edition.pdf.
8. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons: Chichester, UK; 2007.
9. Garfinkel SL. Carving contiguous and fragmented files with fast object validation. Digital Investigation. 2007;4:2–12. DOI: 10.1016/j.diin.2007.06.017.

10. Richard III GG, Roussev V. Scalpel: A frugal, high performance file carver. InDFRWS; 2005.
11. Whitney L. How an 8-character password could be cracked in just a few minutes. TechRepublic. [online] Available from: https://www.techrepublic.com/article/how-an-8-character-password-could-be-cracked-in-less-than-an-hour/. 2023.
12. Datar M, Muthukrishnan S. Estimating rarity and similarity over data stream windows. In: Algorithms—ESA 2002. Proceedings: 10th Annual European Symposium Rome, Italy. Vol. 10. Springer: Berlin, Heidelberg; 2002. pp. 323–35.
13. Kelseyk. (2023). Best Password Practices to Defend Against Modern Cracking Attacks. Specops Software. [online] Available from: https://specopssoft.com/blog/best-password-practices-to-defend-against-modern-cracking-attacks/
14. Scalpel – A frugal, high performance file carver – Dfrws. Dfrws. [online] Available from: https://dfrws.org/presentation/scalpel-a-frugal-high-performance-file-carver/. 2019.
15. Carvey H, Altheide C. Digital Forensics with Open Source Tools. Elsevier: Amsterdam; 2011.
16. Data corruption: Causes, effects & prevention DataCore. DataCore software. [online] Available from: https://www.datacore.com/glossary/data-corruption/. 2024.
17. Carrier B. File System Forensic Analysis. Addison-Wesley Professional: Pearson, USA Education, Inc.; 2005. ISBN: 0-32-126817-2.
18. Kent K, Chevalier S, Grance T, Dang H. Guide to integrating forensic techniques into incident response recommendations of the National Institute of Standards and Technology. National Institute of Standards and Technology. [online] Available from: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf. 2006.
19. Böhme R, Freiling FC. On metrics and measurements. In: Dependability Metrics; 2008. pp. 7–13. DOI: 10.1007/978-3-540-68947-8_2.
20. Goodrich MT, Tamassia R. Algorithm Design: Foundations, Analysis, and Internet Examples. John Wiley & Sons: Chichester, UK; 2001.
21. Ramkumar V, Balaji SB, Sasidharan B, Vajha M, Krishnan MN, Kumar PV. Codes for distributed storage. Foundations and Trends® in Communications and Information Theory. 2022;19:547–813. DOI: 10.1561/0100000115.

